

Cellular Automaton-Based Sentiment Analysis Using Deep Learning Methods

**Elizabeth M. J.
Raju Hazari**

*Department of Computer Science and Engineering
National Institute of Technology, Calicut
Kerala, India*

The swift expansion of internet-centric applications, including social media platforms, online marketplaces and blogs, has given rise to comments, experiences, sentiments, evaluations and reviews including daily life events. Sentiment analysis involves the collection of thoughts, opinions, experiences and impressions on diverse areas, including product reviews on e-market sites, movie reviews and experiences on various e-services. The existing solutions encounter various challenges in effectively managing such diverse and nuanced data. So we present a new approach for classifying such sentiments using the cellular automata-based word vectorization method with deep learning techniques such as long short-term memory (LSTM), convolutional neural network (CNN) and CNN-LSTM models. The designed model makes use of pertinent properties of the signals obtained from the cellular automaton to classify sentiments as positive or negative. We consider a grid of cells governed by a synchronous update based on Wolfram's rules. The cellular automaton evolution process slowly learns the context of the text and reaches a stable state by generating cycle length (CL) signals, after achieving convergence. These signals are used for creating a CL weight matrix. The CL weight matrix is then back-propagated to classify sentiments using standard deep learning methods. The proposed system is a new automated method for assessing semantic linkages and the meaning of the text for natural language applications. LSTM performs well with CL signal weights and produces an accuracy level of 99.18% and an F1_score of 98.68% when trained with sentiment tweets. The experimental results illustrate that the suggested approach has the potential to accurately classify social media comments into different classes and consistently outperforms other baseline models, achieving overall good results.

Keywords: cellular automata; sentiment analysis; machine learning; text vectorization; deep learning

1. Introduction

Sentiment analysis is a recent and existing problem that can be solved with natural language processing (NLP) techniques. The process

automatically determines whether a text expresses positive, negative or neutral opinions. Movie reviews, restaurant reviews, customer feedback, Twitter comment analysis and other sources are some areas where this analysis can be applied. Sentiment analysis is more valuable for people who want to analyze product reviews before purchasing a product as well as for corporations or companies that want to track public perception of their brands or for analyzing movie reviews. Different application areas of sentiment analysis are social media monitoring, analyzing customer feedback, depression analysis, online market management, customer support management and the study of reviews and ratings. Nowadays, it is very important to keep track of what customers believe, to predict future sales and trends and to analyze human behavior. There are two ways to provide feedback on online platforms [1]: Customers can rate their experiences on a scale from 1 (unsatisfactory) to 5 (excellent) by providing their feedback in the form of stars or they can write feedback in the form of in-depth product reviews using textual data and visuals.

In our study, we mainly focused on how to process sentiment analysis tasks without using any syntax, semantics or language structures. We focus on developing a language-independent generic model for processing without using pretrained language models. For the analysis, four datasets were employed, each subdivided into training and testing cases in an 80:20 ratio. Then we applied four different types of deep learning (DL) algorithms, such as dense neural architecture, long short-term memory (LSTM), convolutional neural network (CNN) and LSTM-CNN algorithms. LSTM demonstrates optimal performance across all four datasets, namely restaurant reviews, sentiment tweets, Amazon Alexa reviews and SST-binary movie reviews, yielding notably good F1 scores. The results are shown in Section 5.

The proposed cellular automaton (CA)-based model uses vectorized text as input. The vectorization procedure is done using ASCII values of the characters present in the text and CA module. We will get a set of values called cycle length values for each word in the data file at the end of this evolution process of the CA. These values are used for calculating weights for each word. The same thing is done for each sentence in the dataset to find sentence-level weightage. These weights are used for text-to-vector conversion. Finally, DL models are used for processing the vectorized data. The final output will be the class labels for each sentiment in the data file. The architecture of the proposed model for sentiment analysis, as shown in Figure 1, takes rule vectors as input and predicts class labels as output.

Here are the primary contributions of this study.

1. We propose a new text vectorization method that utilizes the cycle length values of cellular automata (CAs). This method transforms each text sample in the dataset into one-dimensional vectors with the help of CA to learn the context of the text.
2. We describe a systematic DL framework using CA-based text vectors for predicting sentiments associated with social media comments or opinions from online forums. The framework encompasses CNN, dense models, LSTM and Bi-LSTM models.
3. The experimental findings demonstrate that our model outperforms several other models already in use. The best DL model for the CA-based text vectorization technique is LSTM, which produces comparatively good results.
4. We detect and uncover how cellular automata (CAs) play a significant role in the field of NLP as primary research.

Our findings emphasize the need to include public opinion and proper computational methods in comprehending the positive and negative perspectives on human behavior challenges and guiding related decision-making. The following is the general structure of the paper. In Section 2, we give a deep study of previous work carried out on sentiment analysis. Then we give a quick overview of CA processing in Section 3. We delineate the processing methods proposed in our research and the DL techniques applied for sentiment analysis in Section 4. Next, we present the findings, results and discussion about the proposed method in Section 5. Finally, we conclude and discuss future work for analyzing online platforms.

2. Related Study

During the literature study, we can observe a variety of approaches for addressing sentiment analysis tasks. In 2021, a tertiary study was published by Lighthart et al. [2], in which they emphasized that the most popular DL techniques adopted for sentiment analysis are CNN and LSTM. In 2022, a survey was conducted by Wankhade et al. [3] to analyze the challenges of sentiment analysis, and they also provided a complete overview of the methods used for sentiment analysis. Sentiment analysis can be done in different ways, such as document-level sentiment analysis [4, 5], sentence-based sentiment analysis [6, 7] and aspect-based sentiment analysis (ABSA) [8, 9]. Document-level sentiment analysis expresses an opinion on the whole text as positive, negative or neutral. Identifying the general view of the document is the task in this case. In sentence-level sentiment analysis, the objective is to ascertain whether each sentence conveys an opinion

and subsequently determine the overall polarity of that viewpoint as a whole [6, 10].

A lexicon-based dictionary needs to be developed to perform lexicon-based sentiment analysis [11]. The Senti-N-Gram lexicon [12] method and computing the semantic similarity measure between lexicon words and input words [13] are two new methods used for analysis. Aspect extraction and aspect-based sentiment classification are the two tasks associated with aspect-based sentiment analysis [8, 14], which analyzes how people feel about various aspects, entities, people, events, features, objects and specific targets. Nowadays, the bidirectional encoder representations from transformers (BERT)-LSTM [15], CNN with LSTM and CNN with Bi-LSTM [16] are some hybrid models used to enhance the performance of downstream tasks of NLP.

In [8], the authors proposed a generative probabilistic topic modeling approach using a bag of opinion pairs along with a basic LDA model. Their work mainly focused on identifying semantic aspects and aspect-level sentiments from the review and then predicting sentiment labels. Their model achieves an accuracy of 82.26% with 15 aspects on the CD review dataset. In [17], ABSA is carried out in two ways: one is by extracting aspect-based terms from the text. The second method is to classify sentences based on their aspects. Zhao et al. [18] used word2vec embedding for text vectorization and investigated capsule networks with dynamic routing for text classification. Kim [19] also worked on dynamic routing using pretrained Glove word vectors.

To our knowledge, the majority of the existing methods capture the hidden semantic structure of text data and use pretrained language models for classifying sentiments. In recent years, transformer-based models have been widely used for processing sentiment analysis, especially different variations of the BERT model. Pretrained language models (PLMs) undergo self-supervised training on large-scale corpora, but their performance diminishes when fine-tuned on smaller datasets. Jiang et al. [5] used a transfer learning technique with an SST binary dataset. They used the proposed SMART architecture with BERT, RoBERTa, MT-DNN and HNN language models for sentiment analysis. Their SMART with RoBERTa model produces an accuracy of 97.5%. T5-3B [20], MUPPET Roberta Large [21] and ALBERT [22] are other transformer-based models with accuracy of 97.4%, 97.4% and 97.1%, respectively.

The challenges identified in the literature are summarized in Table 1.

Literature	Challenges
A review on sentiment analysis from social media platforms (2023) [23]	How to evaluate the tradeoffs (such as power consumption, cost, computing resources, inference time, ease of development and operational complexity) related to using complex systems like huge language models (BERT and GPT-3) for sentiment analysis?
A survey on ABSA: tasks, methods and challenges (2022) [24]	How to process and develop efficient multi-modal ABSA if reviews and tweets are in different modalities (texts and images)? How to develop a unified model with continual learning to tackle multiple ABSA tasks?
Issues and challenges of ABSA: a comprehensive survey (2022) [25]	Which mechanism can be used to improve aspect extraction? How to improve classification accuracy at the aspect level?
Applications and enhancement of document-based sentiment analysis in DL methods: systematic literature review (2022) [26]	DL algorithms are still insufficient for document-level sentiment analysis. There are several scenarios where the performance of DL models is less effective. How do we analyze the judgment's feeling depending on the feeling of the connected sentence?
A survey on DL for textual emotion analysis in social networks (2022) [27]	There is no balanced corpus for multi-language emotional analysis. The existing corpus resources are for a single-language emotion analysis model that cannot be applied to multi-language emotion analysis. How to do cross-domain emotion analysis using emotional corpus resources of different domains?
A comprehensive survey on sentiment analysis (2021) [28]	What are the mechanisms for creating datasets and lexicons for languages other than English?
Document-level sentiment analysis: a survey (2018) [10]	Sentence-level sentiment analysis is better than document-level analysis. How do we apply sentence-level finer-grained analysis for languages other than English at the aspect level?
A survey on sentiment analysis methods and approach [29]	How to address polarity shift problem in sentiment analysis?

Table 1. An overview of existing surveys.

2.1 Study Related to Vectorization Techniques in Natural Language Processing

Here, we provide an overview of word-oriented representation learning strategies. The two basic forms of word embeddings are [30]:

Static word embedding mechanisms:

- Word2vec. A two-layer neural network model [31] used to convert words to vectors of real numbers. There are two variants:
- Continuous bag-of-words (CBOW). Faster than skip-gram for predicting a target word based on the given context and best model for representing more frequent words.
- Skip-gram model. An unsupervised learning technique for finding most related words of a given word.
- A word2vec model seeks to optimize the average log probability of the context of each word.
- FastText. An open source word embedding model [32] handles out-of-vocabulary (OOV) words. So it requires more memory.
- GloVe. Global vectors model [33] is an unsupervised learning algorithm that determines the semantic links between words. It does not solely rely on local statistics but uses a word co-occurrence matrix for recording co-occurred words to generate the vectors.

Contextual word embedding mechanisms:

- ELMo. A word embedding mechanism [34] that uses a pretrained deep bidirectional language model in which a word can have many vector representations based on the context.
- BERT. A word embedding technique [35] that allows the word vector representation depending on its meaning obtained from its context.
- Generative pretrained transformer (GPT). A pretrained language model that uses 175 billion parameters for text processing [36].

Table 2 presents a comparison of existing word representation learning methods.

Embedding	Algorithm	Language	Task
word2vec [31] by Google (2013)	FFNN, LR	multilingual	unsupervised
BWEs [27] by Stanford University (2013)	neural network	Chinese and English	unsupervised
Polyglot [37] by Stony Brook University (2013)	distributed word representations	multilingual	unsupervised
Glove [33] by Stanford University (2014)	weighted least squares, LR	multilingual	unsupervised
BiDRL [38] by Peking University (2016)	LR	cross-language	semi-supervised
Emoji2Vec [39] by Princeton University and University College London (2016)	LR	English	supervised

SSWE [40] by Harbin Institute of Technology et al. (2016)	FFNN	English	supervised
fastText [32] in 2016 by Facebook	probability statistics	multilingual	supervised
Context2vec [41] by Barilan University (2016)	Bi-LSTM	multilingual	unsupervised
CoVe [42] in 2017	LSTM	English and German	supervised
Emo2Vec [43] in 2018 by HKU of Science and Technology	CNN	English	supervised
ULMFit [44] in 2018 by the University of San Francisco et al.	LSTM	multilingual	supervised, semi-supervised
NTUA-SLP [45] in 2018 by National Technical University of Athens et al.	Bi-LSTM	multilingual	unsupervised
SVD-NS [46] in 2018 by Dalhousie University	SVD	English	unsupervised
cw2vec [47] in 2018 by AFSG et al.	stroke n-grams	Chinese	unsupervised
ELMo [34] in 2018	Bi-LSTMs	multilingual	unsupervised
BERT [35] in 2018 by Google	transformer	multilingual	unsupervised
MNLM [48] in 2019 by NIST et al.	LSTM	multilingual	unsupervised
XLMS in 2019 by Facebook [49]	transformer	cross-lingual	supervised and unsupervised
GPTs [50] by OpenAI (2021)	transformer	multilingual	unsupervised
MACEDONIZER in 2022 [51]	transformer	Macedonian	–

Table 2. Comparison of existing word representation learning methods.

3. Introduction to Cellular Automata

John von Neumann began his exploration of CAs by modeling a biological self-reproduction mechanism. Later, researchers used CAs for different applications such as physical system modeling, random number generation and cryptography [52–55]. These systems operate on a standardized grid of cells and involve local interactions among the

closest adjacent cells, establishing the state of each cell at different time intervals. The state of a cell is influenced by the states of its adjacent cells, governed by predefined local transition rules.

The most basic type of CA, called an elementary CA (ECA), was first suggested by Wolfram [56] in 1986, in which each CA cell records a binary state at a time t . The cell state at the time $t + 1$ is determined by its present state and the current states of its left and right neighboring cells. The next state of a cell is calculated through localized interactions that can be defined as follows:

$$(NS)_i^{t+1} = f((PS)_{i-1}^t, (PS)_i^t, (PS)_{i+1}^t) \quad (1)$$

where f is the transition function used for local interactions of the cells,

- $(NS)_i^{t+1}$ is the next state of the i^{th} cell,
- $(PS)_{i-1}^t$ is the present state of the left neighboring cell of the i^{th} cell,
- $(PS)_i^t$ is the present state of the i^{th} cell and
- $(PS)_{i+1}^t$ is the present state of the right neighboring cell of the i^{th} cell at time t .

The lookup table (see Table 3) of the interactions can be expressed as a function $f: \{0, 1\}^3 \mapsto \{0, 1\}$. There are $2^8 = 256$ ECA rules [56]. Two such rules, 30 and 220, are shown in Table 3.

Present state at t :	111	110	101	100	011	010	001	000	
	(7)	(6)	(5)	(4)	(3)	(2)	(1)	(0)	Rule
state at $t + 1$:	0	0	0	1	1	1	1	0	30
state at $t + 1$:	1	1	0	1	1	1	0	0	220

Table 3. Lookup table for rules 30 and 220.

Traditionally, each of the cells of a CA follows the same next-state function. Such a CA is called a uniform CA. On the other hand, if the CA cells are allowed to follow different next-state functions (rules), the CA is a nonuniform (or hybrid) CA. In this paper, we employed periodic boundary conditions with nonuniform ECAs where the first and last cells are neighbors.

4. Proposed Methodology

The proposed method is called a CA-based machine learning (CAML) model. The preprocessed data is used for the generation of word vectors with the help of CAs. There are three steps:

- 1. Data preprocessing
- 2. Generation of word vectors using cycle length (CL) values
- 3. Apply DL models for classification

4.1 Data Preprocessing

The raw dataset that is gathered is at first cleaned up to remove unwanted characters, special symbols and extra spaces. Now the cleaned data is considered as the reviews. We also extract all the unique words in the preprocessed data and store them in a unique word file. So all unnecessary things need to be removed to accelerate the learning process, and it also helps to reduce memory usage. The algorithm used for data preprocessing is given in Algorithm 1. Some sample reviews of the SST_binary dataset before preprocessing and after preprocessing are given in Table 4.

Before Preprocessing	After Preprocessing
“bÃ©art and berling are both superb , while huppert ... is magnificent .”	“bart and berling are both superb while huppert is magnificent”
“it ’s not too fast and not too slow ”	“it is not too fast and not too slow”
“ ... a fascinating curiosity piece – fascinating , that is , for about ten minutes. ”	“fascinating curiosity piece that is for about ten minutes”
“formula 51 ##### promises a new kind of high but delivers the same old bad trip .”	“formula promises new kind of high but delivers the same old bad trip”
“ I ’ll go out On a Limb.....”	“will go out on limb”

Table 4. Sample reviews before and after employing preprocessing algorithm.

Algorithm 1. Algorithm for data preprocessing.

INPUT. Textual Comments

OUTPUT. Preprocessed comments.

- 1. Convert all text to either uppercase or lowercase, ensuring data integrity.
- 2. Eliminate meaningless words, single characters, numerical expressions, unnecessary spaces, special symbols, non-ASCII characters, punctuation marks, URLs, long repeated patterns and emojis from the text.
- 3. Standardize the text by expanding contractions.
- 4. Removal of user tags: Replace the “@names” with “user” and eliminate unnecessary hashtags for model learning [57].
- 5. Eliminate blank rows and columns from the dataset [58].

4.2 Proposed Cellular Automaton–Based Machine Learning Model

The proposed CAML model for sentiment analysis is shown in Figure 1, which takes rule vectors as input and predicts class labels as output. The preprocessed data file as well as the extracted unique words is converted into rule vectors using ASCII values (see Table 5).

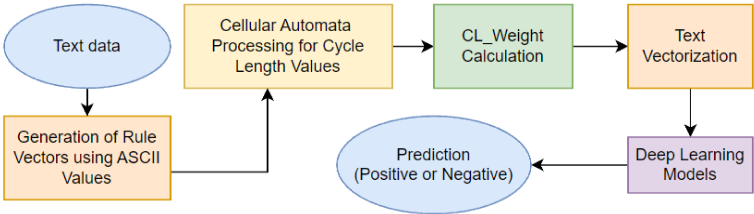


Figure 1. The architecture of the proposed model for sentiment analysis.

Characters	Rule	Character	Rule	Character	Rule
Space	32	h	104	r	114
,	39	i	105	s	115
.	46	j	106	t	116
a	97	k	107	u	117
b	98	l	108	v	118
c	99	m	109	w	119
d	100	n	110	x	120
e	101	o	111	y	121
f	102	p	112	z	122
g	103	q	113		

Table 5. Rule table used for CA evolution.

For example,

- Text-1: “the film is strictly routine”
- Rule vectors = [116, 104, 101, 32, 102, 105, 108, 109, 32, 105, 115, 32, 115, 116, 114, 105, 99, 116, 108, 121, 32, 114, 111, 117, 116, 105, 110, 101].

These rule vectors are given to a CA for evolution. The cells considered for CAs are filled with alternate zeros and ones. We can consider any initial configuration, but we have to use the same initial configuration for all cases. The number of cells in the CA equals the size of the text. Using the rules and the states of the neighborhood cells, each cell in the CA machine evolves with a finite set of values in discrete time. In ECA, every CA cell saves a binary state at time t . The subsequent state is determined by considering its current state along with the

states of its two immediate neighbors. In this paper, the CA evolves 1000 times if the number of cells is less than 2000. If the number of cells is greater than 2000, the evolution value is taken as half of the size of the cells because, in each iteration, the effect of a cell is passed to its two nearest neighborhood cells through periodic boundary conditions. We observe that after a certain number of iterations on a particular cell, the pattern gets repeated (see Figure 2).

If a pattern is repeated at least 32 times, its length is called the CL value (since CL numbers represent the size of the repeating pattern, we may substitute any value for 32 and the performance will not be impacted with this value). If the CA cell has no such repeating patterns, the CL value is set to -5 . We can also use zero or any other negative integer instead of -5 . The evolution of a CA is shown in Figure 2. Bold boxes in the figure indicate the patterns that reoccur after specific evolutionary steps. These patterns can be of any length. In column 2 of Figure 2, the string 011 starts to repeat; its length is three, representing the cycle length value of the cell. While the sequence 01 begins repeating from the third row of the fifth column, the CL value for that cell will be 2. Similarly, we get the CL values for each CA cell. Each character in the text is converted into rule vectors using Table 5. These vectors are then utilized to generate CL values. The collection of CL values produced during the CA evolution process for the given text is referred to as the CL sequence.

.....	1	0	1	0	1	0	1	0	
.....	0	1	0	1	0	1	1	0	
.....	1	0	1	0	1	1	1	0	
.....	1	0	0	1	0	0	1	1	
.....	0	0	1	0	0	1	1	1	
.....	1	1	0	1	1	0	1	0	
.....	1	1	0	0	1	0	0	0	
.....	0	0	0	1	1	0	0	1	
.....	1	1	0	0	0	1	1	0	
.....	1	0	0	1	1	0	0	0	
.....	0	1	1	0	0	0	0	0	
.....	1	0	0	1	0	0	0	0	
.....	1	0	0	0	0	1	0	1	
.....	:	:	:	:	:	:	:	:	

Figure 2. CA evolution process.

4.3 Generation of Word Vectors Using Cycle Length Values

As discussed earlier, the proposed model extracts each unique word from the dataset and generates CL values for each word. The CL

values of sentences and the CL value of each word are used to calculate the CL weight of any particular word. The CL weight calculation procedure is as follows:

- Sentence CL sum (X): It is the sum of the CL values of the sentence.
- Word CL sum (Y): It is the sum of the CL values of the word.
- CL weight: Let us consider that the word α is present in n number of sentences in the dataset and the word CL sum of α is Y_α ; then CL weight of α is calculated by:

$$\frac{\sum_{i=1}^n X_i}{Y_\alpha}.$$

For example, let us consider a dataset with only two sentences; then, the word vectorization is done using the CL weight of each word in the sentence, shown in Table 6.

- Sentence 1: “the film is strictly routine”
- CL values = [3, 3, 1, 1, 1, 1, 1, 1, 1, 6, 6, 6, 2, 16, 16, 16, 8, 8, 8, 8, 8, 2, 4, 4, 4, 4, 1, 1]
Sentence CL sum = 141
- Sentence 2: “the drama discloses almost nothing”
- CL values = [3, 3, 1, 1, 1, 3, 3, 3, 3, 1, 1, 1, 1, 1, 1, 3, 3, 3, 3, 1, 1, 1, 11, 11, 11, 11, 11, 11, 11, 1, 1, 1, 1, 1]
Sentence CL sum = 122

Unique Words	CL Values	Word CL Sum	CL Weight
the	[1, 1, 1]	3	$(141 + 122)/3 = 87$
film	[1, 1, 1, 1]	4	$141/4 = 35$
is	[1, 1]	2	$141/2 = 70$
strictly	[3, 3, 3, 3, 3, 3, 3, 3]	24	$141/24 = 5$
routine	[12, 12, 3, 12, 12, 12, 12]	75	$141/75 = 1$
drama	[1, 1, 1, 1, 1]	5	$122/5 = 24$
discloses	[6, 6, 6, 6, 6, 6, 3, 3, 3]	45	$122/45 = 2$
almost	[3, 1, 3, 3, 3, 3]	16	$122/16 = 7$
nothing	[1, 1, 1, 1, 4, 4, 2]	14	$122/14 = 8$

Table 6. Calculation of CL weight of each unique word in the dataset.

If we take the CL weight of each word of a sentence in the same order, then we get the weight vector for the particular sentence. That is, we get a vector for each sentence in the dataset represented in Table 7. These vectors are given as input to the DL models for predicting the class labels.

Reviews	Vectors
“the film is strictly routine”	[87, 35, 70, 5, 1]
“the drama discloses almost nothing”	[87, 24, 2, 7, 8]

Table 7. Sentence to vector conversion.

4.4 Deep Learning Methods Used for Processing

We have used five different types of DL models for processing our data: LSTM, Bi-LSTM, CNN, dense architecture and CNN-LSTM. The dataset in vector form is used as input for the models. The details of each model and the algorithm used for prediction are discussed in the following section.

4.4.1 Long Short-Term Memory

LSTM is a DL neural network for learning long-term dependencies of data. It is an artificial recurrent neural network (RNN) that has feed-back connections to process entire sequences of vectors in the data. Therefore, it is mainly used for the processing of time series data. The basic cell structure of LSTM is shown in Figure 3.

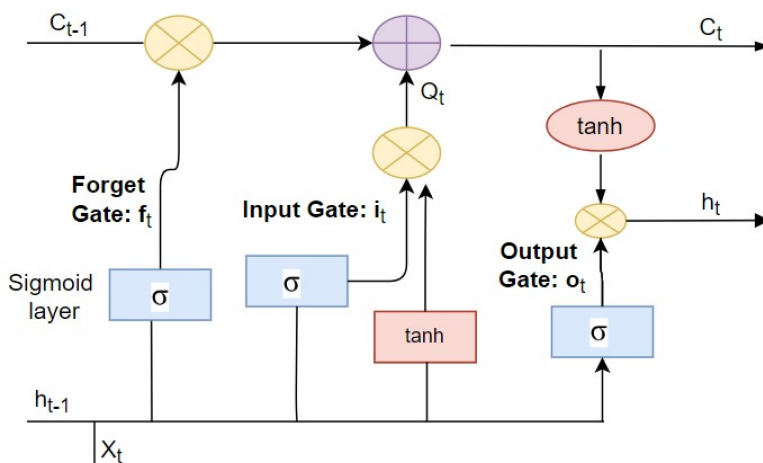


Figure 3. Basic cell structure of LSTM.

An input gate, an output gate and a forget gate make up a single LSTM unit. These gates regulate the information flow into and out of the cell. If the output of the forget gate is 1, then the information is thoroughly taken into consideration, and if it is a 0, then that information is completely ignored. The input gate layer of LSTM determines which values will be updated first. The output layer helps to output

only the information that we need. The processing of these states is described mathematically in equations (2)–(4). The current state of the cell is calculated using equations (5) and (6). The state of the hidden cell is obtained using equation (7).

Input gate:

$$i_t = \sigma(W_i(X_t, h_{t-1}) + b_i) \quad (2)$$

Forget gate:

$$f_t = \sigma(W_f(X_t, h_{t-1}) + b_f) \quad (3)$$

Output gate:

$$o_t = \sigma(W_o(X_t, h_{t-1}) + b_o) \quad (4)$$

Memory cell:

$$Q_t = i_t * \tanh(X_t) \quad (5)$$

$$C_t = f_t * C_{t-1} + Q_t \quad (6)$$

Result:

$$h_t = o_t * \tanh(C_t) \quad (7)$$

where X_t denotes the CA-based word vector at time t . The f_t , i_t and o_t represent the output values (activation vectors) of the input gate, forget gate and output gate at the time t , respectively. C_t is the memory unit vector. h_t represents the output vector of the LSTM unit at a time t . The self-updating weight of the hidden layer is denoted by W , and the term b denotes the bias vector. The sigmoid function and hyperbolic tangent functions are activation functions, represented by using $\sigma(\cdot)$ and $\tanh(\cdot)$, respectively. The operator $*$ denotes elementwise multiplication. All the hidden layer outputs and gate values lie within the range of $[0, 1]$.

Figure 4 depicts the architecture of our LSTM sequential model, encompassing five layers: the input layer, embedding layer, LSTM layer, dropout layer and dense layer. The embedding layer executes embedding operations on input data, while the dropout layer addresses overfitting issues by filtering out unwanted characters and symbols, commonly referred to as noise. This helps in removing noisy data and preventing the model from overfitting. The standard deeply linked neural network layer is called the dense layer. It is the layer that is most often utilized and prevalent. The number of neurons defined inside the layer determines the number of output class labels. Our task is a predictive modeling problem in which every sequence is a variable sequence of words, and each word is represented by cycle-length weight. To learn the long-term context of the input sequence, we used an Adam optimization algorithm that dynamically adjusts the learning rates of the model.

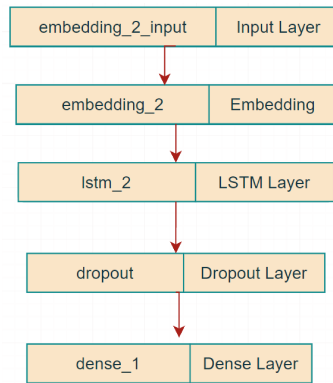


Figure 4. LSTM sequential model with five different layers.

4.4.2 Convolutional Neural Network

CNN is a DL technique for processing data with geographical or temporal relationships that contains a series of layers such as the convolutional layer (which is made up of a set of filters), pooling layer (for reducing the size of the feature map) and fully connected layers (used at the end of a CNN). Shervin et al. presented a CNN model in [59], in which words are stacked to create a two-dimensional array representing a phrase or paragraph. Next, they utilized CNN with softmax activation on the SST-Binary dataset and IMDB movie reviews. In this paper, we employed one-dimensional convolutional neural networks (CNNs) with the rectified linear unit (ReLU) activation function. The extraction of features was facilitated by 64 filters, each with a kernel size of 32. The processing steps are shown in Figure 5.

4.4.3 Convolutional Neural Network–Long Short-Term Memory Model

The CNN-LSTM model is a combination of CNN and LSTM models. In [60] Gandhi et al. proposed CNN and LSTM to classify positive and negative movie reviews from the IMDB dataset. Each review in the dataset is converted into vectors with the help of word2vec embedding. Their model produces the validated tweet accuracy of 88.02% with the LSTM model and testing accuracy of 87.72% with the CNN model using the IMDB dataset. A combination of the CNN-LSTM model is used in [61] for predicting the valence arousal (VA) ratings of texts. The CNN-LSTM model we used is shown in Figure 6, in which the embedding layer uses an embedding size of 256 to convert input sequences of integers into dense vectors. A one-dimensional convolutional layer is used to extract 256 features from the input sequences. Dropout layers are used to prevent overfitting and LSTM layers are for learning long-term dependencies in the sequence data.

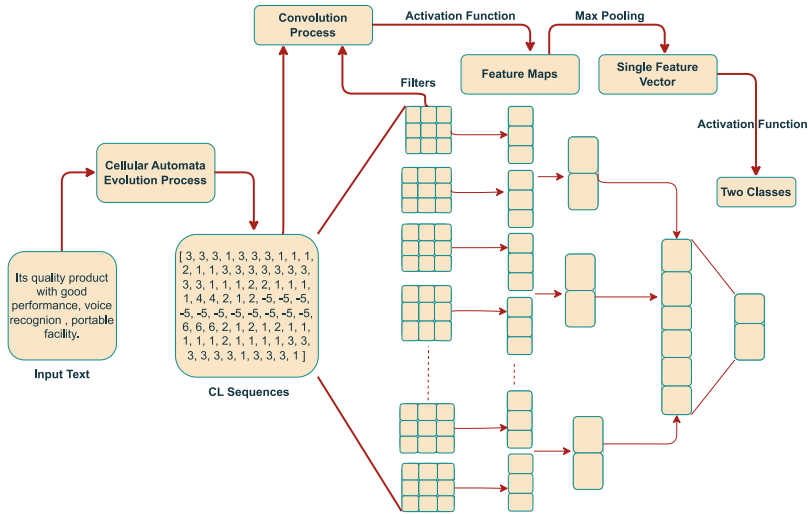


Figure 5. The CA-based CNN model for the sentiment analysis.

4.4.4 Dense Architecture

Dense architecture is a simple layer of neurons that produces an output using the following operation:

$$\text{activation}(\text{dot}(\text{input}, \text{kernel}) + \text{bias}) \quad (8)$$

where activation is the activation function. We used the sigmoid function to introduce nonlinearity into the model for learning decision boundaries. The layer creates a vector called bias and a weight matrix called kernel to optimize the model and modify the dimensionality of the output from the previous layer. The dot operator is used with all input and its corresponding weights.

Each of the four DL models demonstrated satisfactory performance when utilizing CL weight-based text vectors. Notably, they achieved this without relying on linguistic features such as absolute word positions, parts of speech tags, semantic lexicons or any pretrained language models. Algorithm 2 elucidates the comprehensive step-by-step process employed in this approach.

Algorithm 2. Algorithm for sentiment analysis using CA-based DL models.

INPUT. Textual comments

OUTPUT. Predicted labels for each comment.

1. Preprocess each comment in the dataset for generating rule vectors using Algorithm 1.
2. Convert cleaned reviews/comments into CA rule vectors using Table 5.

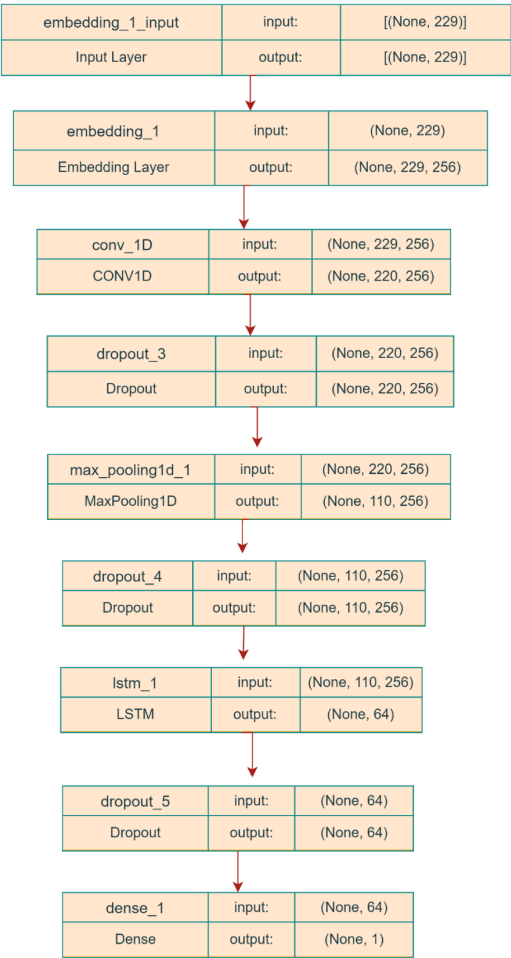


Figure 6. The CNN-LSTM model used for the prediction of Amazon Alexa reviews.

3. Identify CL sequences using CA rule vectors.
4. Calculate CL weight for each unique word in the dataset.
5. Perform text vectorization process.
6. Implement a sequence of LSTM/Bi-LSTM/CNN/Dense/CNN-LSTM models for sentiment analysis using the Adam optimizer and ReLU (at intermediate layer) and sigmoid (at output layer) activation functions.
7. Train the model using the training dataset.
8. Predict the output labels of testing data.

9. Find precision, recall, accuracy and F1_Score using the following formulas for analyzing the performance of the proposed approach:

$$P = TP / (TP + FP) \quad (9)$$

$$R = TP / (TP + FN) \quad (10)$$

$$\text{Accuracy} = (TP + TN) / \text{total number of sentiments} \quad (11)$$

$$\text{F1_Score} = (2 * P * R) / (P + R) \quad (12)$$

where P stands for precision and R is for recall.

5. Results and Performance Analysis

In this section, we give the dataset descriptions, hyperparameters used for fine-tuning and the results obtained for various DL models. The comparative analysis is also mentioned in detail.

5.1 Datasets Description

The study involved collecting datasets from various sources, including restaurant reviews, Twitter sentiments, SST-binary classification and Amazon Alexa reviews. A detailed description of each dataset follows.

- Restaurant Reviews [62] had 1000 reviews mixed with 500 positive and 500 negative sentiments.
- Twitter Sentiment [63] contains 10 715 tweets distributed into 8402 positive and 2313 negative reviews.
- SST-Binary [64] has 9613 movie reviews with 4963 positive and 4650 negative labels for sentiment classification.
- Amazon Alexa Reviews [65] comprises roughly 3000 Amazon customer reviews of different Amazon Alexa goods including Alexa Echo, Echo dots and Alexa Firesticks. This unbalanced dataset contains 2828 positive reviews and 238 negative reviews.

5.2 Hyperparameter Tuning

The sequential network models are trained over 50 epochs with the validation split of 0.2 using a back-propagation technique—the Adam stochastic optimization approach with a learning rate of 0.0005 and patience of 5. The mode of verbosity is defined as 2. An embedding layer is used as the first hidden layer for all models with an input dimension of 10 001 (size of the vocabulary in the text data) and an output dimension of 16. The `binary_crossentropy` function is employed to calculate the loss between the true and predicted classes. Following each iteration, the network undergoes testing on validation data to detect convergence. The hyperparameter details follow.

- **Dense Architecture.** The model incorporates two TensorFlow dense layers. The initial dense layer comprises 24 neurons, receiving the input with a ReLU activation function. The output layer is also a dense layer activated using a sigmoid function. Additionally, a one-dimensional global average pooling layer is introduced to downsample the input features. For regularization, a dropout layer with a value of 0.2 is also applied.
- **Bi_LSTM Model.** A one-dimensional convolution layer is employed to process the input, utilizing ReLU as the activation function. This bi-directional LSTM model comprises three LSTM layers. The initial LSTM layer is configured with 128 neurons, a dropout value of 0.5 and a recurrent dropout of 0.5. The subsequent two LSTM layers consist of 64 neurons each, with a dropout value of 0.4. The output layer incorporates a dense layer with a sigmoid activation function.
- **LSTM Model.** There are three LSTM layers. The first layer contains 512 neurons with a dropout of 0.2. The second layer contains 128 neurons with a dropout of 0.2, and the third layer has only 64 neurons. A softmax activation function is used in the output layer.
- **CNN Model.** The model incorporates a one-dimensional convolution layer with 64 filters and a kernel size of 32, receiving the input with a ReLU as an activation function. To downsample the input features, a one-dimensional global average pooling layer with a pool size of 2 is introduced. Following this, a flattened layer is employed to flatten the input before reaching the output layer, and a dropout of 0.3 is applied for network regularization. Before the output layer, a dense layer comprising 16 neurons with a ReLU activation function is included. The output dense layer utilizes the sigmoid activation function.
- **CNN with LSTM Model.** The model employs a one-dimensional convolution layer with 256 neurons, taking the input with a ReLU as an activation function and incorporating a dropout of 0.3. A one-dimensional global average pooling layer is introduced with a pool size of 2. Subsequently, a flattened layer is applied to flatten the data, utilizing the same dropout value. Three LSTM layers follow, with the first layer having 128 neurons and the subsequent two layers having 64 neurons, all with a recurrent dropout of 0.5.

■ 5.3 Matrices Used

The details of the matrices used are illustrated in Figure 7. A few sample confusion matrices of the proposed model on the Amazon Alexa Review dataset are given in Figure 8. These confusion matrices are used to find out the accuracy, precision, recall and F1 score.

		Predicted		
		Positive	Negative	
Actual	Positive	True Positive (TP)	False Positive (FP) Type I Error	Recall TP/(TP+FN)
	Negative	False Negative (FN) Type II Error	True Negative (TN)	Specificity TN/(TN+FP)
		Precision TP/(TP+FP)	Negative Predictive Value TN/(TN+FN)	Accuracy (TP+TN)/(TP+TN+FP+FN)
F1_Score = 2 x (Precision x Recall) / (Precision + Recall)				

Figure 7. The details of matrices used for our work.

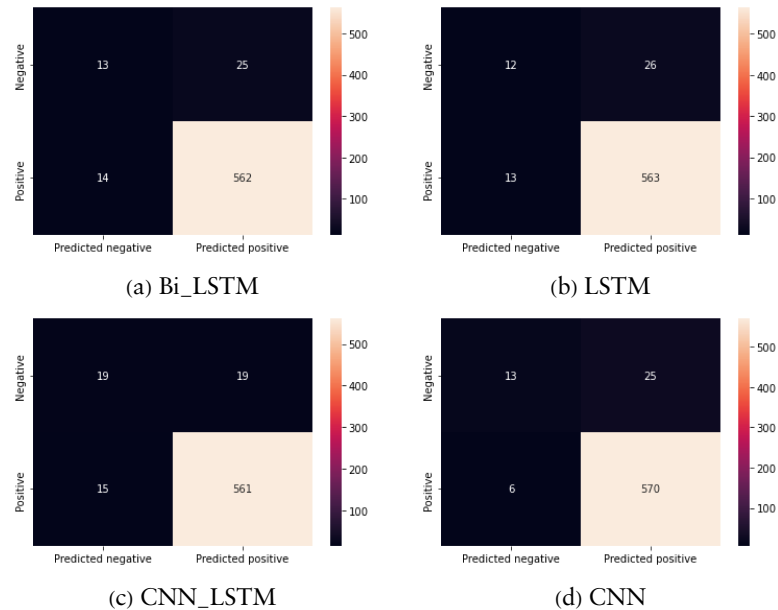


Figure 8. Confusion matrices of various models on the dataset of Amazon Alexa reviews using the proposed methodology (for training, 80% of data is used and for testing, 20% of data is used with proposed methodology).

5.4 Experimental Results

The Spyder Python environment is used for experimental analysis with 80% of training data and 20% of testing data for evaluating the model. The datasets are divided into an 80:20 ratio using Python’s

train_test_split helper function. We have implemented all five DL models mentioned earlier, LSTM, Bi-LSTM, CNN, dense architecture and CNN-LSTM, for processing all four datasets described. The accuracy that we obtained is summarized in Table 8.

We found that the LSTM model performs better than all four other models in terms of accuracy.

Dataset	LSTM	Bi-LSTM	CNN	Dense	CNN-LSTM
sentiment tweets [63]	99.18	98.78	81.62	90.02	98.31
Amazon Alexa [65]	94.93	93.65	94.95	93.81	94.46
restaurant [62]	82.00	78.30	79.30	68.00	80.89
SST-binary 64	81.35	74.74	73.39	77.02	76.61

Table 8. The accuracy obtained for the proposed method with different DL models.

From Table 8, it is evident that the simple LSTM layer outperforms and produces comparatively good accuracy with the CA-based CL weight vectorization method. The precision, recall and F1 score values are also analyzed to evaluate the performance of the model. The results obtained for the LSTM model using all four datasets are given in Table 9.

Dataset	Prec.(%)	Rec.(%)	F1(%)	Acc.(%)
sentiment tweet	98.80	98.88	98.68	99.18
Amazon Alexa	93.06	93.81	93.86	94.93
restaurant review	77.83	77.84	75.75	82.00
SST-binary	81.14	80.16	81.11	81.35

Table 9. Precision, recall, F1 score and accuracy of the proposed method with LSTM model.

5.5 Comparison with Other Research Work

We compare our CA-based DL model with other state-of-art classification approaches. The baselines selected for text classification are based on the datasets used.

- Sentiment Tweets [63]. In the sentiment tweet dataset, Gautam and Yadav [66] proposed a semantic analysis method using WordNet. Their approach involved calculating the similarity of synonyms in the feature list to determine adjective scores and polarity in reviews. Their method achieved an impressive accuracy of 89.9% on Twitter data. Tan et al. [67] researched three distinct datasets: IMDb, Sentiment140 and Twitter US Airline Sentiments, employing a RoBERTa-LSTM model for sentiment analysis. Meanwhile, Singh and Tripathi [68] explored three

models, namely SVM, decision tree and the random forest algorithm, achieving superior performance with the decision tree algorithm. However, they reported relatively lower F1 measures based on 14 000 tweets collected from KFC and McDonald's datasets. The comparative study is given in Table 10.

- Restaurant Reviews [62]. M. Govindarajan [69] proposed a hybrid ensemble comprising naïve Bayes (NB), support vector machine (SVM) and genetic algorithm (GA) for solving sentiment analysis on restaurant reviews. The J48 decision tree classification [70] produces only 45.60% accuracy on real-world data, crawled from the TripAdvisor restaurant website using WebHarvy. Hamad et al. [71] gave a NB algorithm by extracting unigram features. Hossain et al. worked on restaurant reviews and they got an accuracy of 80.48% for 1000 restaurant reviews using the TF-IDF feature extraction technique. When they increased the dataset size to 6k and used a Bi-LSTM deep learning approach along with word2vec, their model outperformed 6625 Bengali restaurant reviews. In [72] Salim Sazzed proposed a hybrid model of opinion mining on 2315 Bangla restaurant reviews with an accuracy of 92.50% by integrating SVM with the lexicon-based method. A comparative study is shown in Table 10.
- SST-Binary [64]. Jiang et.al. [5] proposed an innovative robust-based approach with SST-Binary to solve sentiment analysis. Tang et al. [73] proposed BERT with a distillation approach for transferring knowledge at the output level using softmax activation. Xu et al. [43] proposed an Emo2Vec encoding mechanism that encodes emotional semantics into vectors. The proposed Emo2Vec logistic regression model is used with the GloVe language model, which produces 81.2% accuracy on the SST binary dataset. Chen et al. [74] proposed a dual contrastive learning-based approach on the SST-Binary dataset. They used 7447 data points for training and 1821 for testing. The designed model learns discriminative features and parameters of the classifier. Recent research models are summarized and compared with our model in Table 10.
- Amazon Alexa Reviews [65]. From the experimental findings, it is clear that the model outperforms many state-of-the-art models. What proportion of predictions is accurate may be determined by looking at the confusion matrices (Figure 8). The results of the LSTM model are compared with other baseline models given in Table 10. The CA-based CNN-LSTM model produces good precision, recall and F1 score with an 80:20 ratio of Amazon Alexa reviews. The maximum testing accuracy we obtained is 94.93% with an 80:20 ratio of Amazon Alexa reviews.

6. Conclusion

This paper aims to provide a completely new approach without using any pretrained models for word embeddings. The text vectorization method we utilized is dynamic and operates regardless of the linguis-

tic or semantic characteristics of the data. The main aspect of the proposed approach is the exploration of the role of cellular automata (CAs) in natural language processing (NLP) for tackling downstream problems, as revealed by the findings. The significance of this research lies in its departure from the use of pretrained tools or language models, positioning it at the forefront of advancements in the field. Future endeavors will involve the application of attention mechanisms with CAs to further enhance the system’s performance.

Dataset	Methods Used	Accuracy	Precision	Recall	F1 Score
Sentiment Tweets (10 715 tweets)	RoBERTa-LSTM using 1.6 million tweets (2022) [67]	89.70	90	90	90
	decision tree on 14000 tweets (2021) [68]	88.51	79.62	84.88	55.57
	SVM on 14000 tweets (2021) [68]	87.71	80.66	85.73	73.02
	Bi-CNN-RNN with attention (2021) on airline tweets [75]	92.75	89.44	88.43	88.77
	ABCDM(2021) on 1.6 million tweets [75]	81.82	82.77	82.31	81.99
	domain attention model on Sanders 5513 tweets (2018) [76]	87.69	–	–	–
	GloVe-DCNN on 3326 sentiments of Twitter data (2018) [77]	81.36	80.61	80.86	80.72
	Proposed CA-based LSTM model	99.18	98.8	98.88	98.68
Amazon Alexa (3066 reviews)	BERT with 4K Amazon reviews (2021) [78]	98.4	98.4	98.4	98.4
	Bi_LSTM [78] with 4K Amazon reviews (2021)	97.4	97.4	97.4	97.4
	CNN-Glove (2019) using 21k unbalanced data [79]	92.73	92	92	92
	CNN-Glove (2019) using balanced data [79]	76.51	76	76	76
	domain attention model for multi-domain sentiments of 8000 reviews (books, DVD, electronics, kitchen (2018) [76]	84.38	–	–	–
	Proposed CA-Based LSTM model	94.93	93.06	93.81	93.86

Table 10. (continues).

Dataset	Methods Used	Accuracy	Precision	Recall	F1 Score
Restaurant Data (1000 reviews)	NB on 68 online restaurant reviews (2021) [71]	73.00	68	80.07	73.54
	NB on 337 restaurant customer reviews (2019) [80]	72.06	71.43	73.53	72.46
	text blob on 337 restaurant customer reviews (2019) [80]	69.12	64.15	94.45	76.41
	DT Classifier-J48 (2019) [70]	45.20	48.7	36.8	41.92
	NB Classifier (2019) [81]	72.2	71.29	64.58	67.76
	Proposed CA-Based LSTM model	82.00	75.83	77.84	77.75
SST-Binary (9613 reviews)	DCL with RoBERTa (2022) [74]	94.91	–	–	–
	contextual sentiment embeddings with Bi-GRU (2022) [82]	89.8	–	–	–
	CNN - word2vec (2021) [83]	86.61	–	–	–
	MUPPET Roberta Large (2021) [21]	97.40	–	–	–
	SMART RoBERTa on 67k reviews (2020) [5]	97.50	–	–	–
	BERT with distillation approach (2019) [73]	90.70	–	–	–
	dynamic routing - capsule networks (2018) [18]	86.80	–	–	–
	LR with Emo2vec and GLove (2018) [43]	81.20			
	Proposed CA-Based LSTM model	81.35	81.14	80.16	81.11

Table 10. Comparison of our model with other baseline models. Weighted average values are taken into account using datasets with an 80:20 ratio.

References

[1] U. A. Chauhan, M. T. Afzal, A. Shahid, M. Abdar, M. E. Basiri and X. Zhou, “A Comprehensive Analysis of Adverb Types for Mining User Sentiments on Amazon Product Reviews,” *World Wide Web*, 23, 2020 pp. 1811–1829. doi:10.1007/s11280-020-00785-z.

[2] A. Ligthart, C. Catal and B. Tekinerdogan, “Systematic Reviews in Sentiment Analysis: A Tertiary Study,” *Artificial Intelligence Review*, 54, 2021 pp. 4997–5053. doi:10.1007/s10462-021-09973-3.

[3] M. Wankhade, A. C. S. Rao and C. Kulkarni, “A Survey on Sentiment Analysis Methods, Applications, and Challenges,” *Artificial Intelligence Review*, 55(7), 2022 pp. 5731–5780. doi:10.1007/s10462-022-10144-1.

- [4] V. Ng, S. Dasgupta and S. M. N. Arifin, “Examining the Role of Linguistic Knowledge Sources in the Automatic Identification and Classification of Reviews,” in *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions (COLING/ACL 2006)*, Sydney, Australia (V. Ng, ed.), Stroudsburg, PA: Association for Computational Linguistics, 2006 pp. 611–618.
- [5] H. Jiang, P. He, W. Chen, X. Liu, J. Gao and T. Zhao, “SMART: Robust and Efficient Fine-Tuning for Pre-trained Natural Language Models through Principled Regularized Optimization,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, Online (H. Jiang, ed.), Stroudsburg, PA: Association for Computational Linguistics, 2020 pp. 2177–2190. arxiv.org/abs/1911.03437.
- [6] B. Pang and L. Lee, “A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts,” in *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL 2004)*, USA (B. Pang, ed.), Stroudsburg, PA: Association for Computational Linguistics, 2004 pp. 271–es. doi:10.3115/1218955.1218990.
- [7] J. Su, Q. Chen, Y. Wang, L. Zhang, W. Pan and Z. Li, “Sentence-Level Sentiment Analysis Based on Supervised Gradual Machine Learning,” *Scientific Reports*, **13**, 2023 p. 14500. doi:10.1038/s41598-023-41485-8.
- [8] Z. Hai, G. Cong, K. Chang, P. Cheng and C. Miao, “Analyzing Sentiments in One Go: A Supervised Joint Topic Modeling Approach,” *IEEE Transactions on Knowledge and Data Engineering*, **29**(6), 2017 pp. 1172–1185. doi:10.1109/TKDE.2017.2669027.
- [9] Ł. Augustyniak, T. Kajdanowicz and P. Kazienko, “Comprehensive Analysis of Aspect Term Extraction Methods Using Various Text Embeddings,” *Computer Speech and Language*, **69**, 2021 101217. doi:10.1016/j.csl.2021.101217.
- [10] S. Behdenna, F. Barigou and G. Belalem, “Document Level Sentiment Analysis: A Survey,” *EAI Endorsed Transactions on Context-aware Systems and Applications*, **4**(13), 2018 154339. doi:10.4108/eai.14-3-2018.154339.
- [11] H. S. Hota, D. K. Sharma and N. Verma, “14-Lexicon-Based Sentiment Analysis Using Twitter Data: A Case of Covid-19 Outbreak in India and Abroad,” *Data Science for COVID-19*, 2021 pp. 275–295. doi:10.1016/B978-0-12-824536-1.00015-0.
- [12] A. Dey, M. Jenamani and J. J. Thakkar, “Senti-N-Gram: An n -gram Lexicon for Sentiment Analysis,” *Expert Systems with Applications*, **103**, 2018 pp. 92–105. doi:10.1016/j.eswa.2018.03.004.
- [13] O. Araque, G. Zhu and C. A. Iglesias, “A Semantic Similarity-Based Perspective of Affect Lexicons for Sentiment Analysis,” *Knowledge-Based Systems*, **165**, 2019 pp. 346–359. doi:10.1016/j.knsys.2018.12.005.

- [14] M. H. Phan and P. O. Ogunbona, “Modelling Context and Syntactical Features for Aspect-Based Sentiment Analysis,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, Online (D. Jurafsky, J. Chai, N. Schluter and J. Tetreault, eds.), Stroudsburg, PA: Association for Computational Linguistics, 2020 pp. 3211–3220. doi:10.18653/v1/2020.acl-main.293.
- [15] R. Mokhosi, C. Shikali, Z. Qin and Q. Liu, “Maximal Activation Weighted Memory for Aspect Based Sentiment Analysis,” *Computer Speech and Language*, 76, 2022 101402. doi:10.1016/j.csl.2022.101402.
- [16] K. Shanmugavadivel, S. H. Sampath, P. Nandhakumar, P. Mahalingam, M. Subramanian, P. K. Kumaresan and R. Priyadharshini, “An Analysis of Machine Learning Models for Sentiment Analysis of Tamil Code-Mixed Data,” *Computer Speech & Language*, 76, 2022 101407. doi:10.1016/j.csl.2022.101407.
- [17] H. Luo, T. Li, B. Liu and J. Zhang, “DOER: Dual Cross-Shared RNN for Aspect Term-Polarity Co-extraction,” in *Annual Meeting of the Association for Computational Linguistics (ACL 2019)*, Florence, Italy (A. Korhonen, D. Traum and L. Màrquez, eds.), Stroudsburg, PA: Association for Computational Linguistics, 2019 pp. 591–601. doi:10.18653/v1/P19-1056.
- [18] W. Zhao, J. Ye, M. Yang, Z. Lei, S. Zhang and Z. Zhao, “Investigating Capsule Networks with Dynamic Routing for Text Classification,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, Brussels, Belgium (E. Riloff, D. Chiang, J. Hockenmaier and J. Tsujii, eds.), Stroudsburg, PA: Association for Computational Linguistics, 2018 pp. 3110–3119. doi:10.18653/v1/D18-1350.
- [19] Y. Kim, “Convolutional Neural Networks for Sentence Classification.” arxiv.org/abs/1408.5882v2.
- [20] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li and P. J. Liu, “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer,” *Journal of Machine Learning Research*, 21(1), 2019 pp. 5485–5551. dl.acm.org/doi/10.5555/3455716.3455856.
- [21] A. Aghajanyan, A. Gupta, A. Shrivastava, X. Chen, L. Zettlemoyer and S. Gupta, “Muppet: Massive Multi-task Representations with Pre-finetuning,” in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021)*, Online (M. Moens, X. Huang, L. Specia and S. Yih eds.), Stroudsburg, PA: Association for Computational Linguistics, 2021 pp. 5799–5811. doi:10.18653/v1/2021.emnlp-main.468.

- [22] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma and R. Soricut, "ALBERT: A Lite BERT for Self-Supervised Learning of Language Representations," in *The Eighth International Conference on Learning Representations (ICLR2020)*, Addis Aaba, Ethiopia, Appleton, WI: International Conference on Learning Representations, 2020. openreview.net/forum?id=H1eA7AEtvS.
- [23] M. Rodríguez-Ibáñez, A. Casáñez-Ventura, F. Castejón-Mateos and P.-M. Cuenca-Jimenez, "A Review on Sentiment Analysis from Social Media Platforms," *Expert Systems with Applications*, **223**, 2023 119862. doi:10.1016/j.eswa.2023.119862.
- [24] W. Zhang, X. Li, Y. Deng, L. Bing and W. Lam, "A Survey on Aspect-Based Sentiment Analysis: Tasks, Methods, and Challenges," *IEEE Transactions on Knowledge and Data Engineering*, **35**(11), 2023 pp. 11019–11038. doi:10.1109/TKDE.2022.3230975.
- [25] A. Nazir, Y. Rao, L. Wu and L. Sun, "Issues and Challenges of Aspect-Based Sentiment Analysis: A Comprehensive Survey," *IEEE Transactions on Affective Computing*, **13**(2), 2022 pp. 845–863. doi:10.1109/TAFFC.2020.2970399.
- [26] F. Alshuwaier, A. Areshey and J. Poon, "Applications and Enhancement of Document-Based Sentiment Analysis in Deep Learning Methods: Systematic Literature Review," *Intelligent Systems with Applications*, **15**, 2022 200090. doi:10.1016/j.iswa.2022.200090.
- [27] S. Peng, L. Cao, Y. Zhou, Z. Ouyang, A. Yang, X. Li, W. Jia and S. Yu, "A Survey on Deep Learning for Textual Emotion Analysis in Social Networks," *Digital Communications and Networks*, **8**(5), 2022 pp. 745–762. doi:10.1016/j.dcan.2021.10.003.
- [28] M. Birjali, M. Kasri and A. Beni-Hssane, "A Comprehensive Survey on Sentiment Analysis: Approaches, Challenges and Trends," *Knowledge-Based Systems*, **226**, 2021 107134. doi:10.1016/j.knosys.2021.107134.
- [29] A. M. Abirami and V. Gayathri, "A Survey on Sentiment Analysis Methods and Approach," in *2016 Eighth International Conference on Advanced Computing (ICoAC'16)*, Chennai, India, Piscataway, NJ: Institute of Electronics and Electrical Engineers, 2016 pp. 72–76. doi:10.1109/ICoAC.2017.7951748.
- [30] E. Getzen, Y. Ruan, L. Ungar and Q. Long, "Mining for Health: A Comparison of Word Embedding Methods for Analysis of EHRs Data." *medRxiv*. doi:10.1101/2022.03.05.22271961.
- [31] A. Nematzadeh, S. C. Meylan and T. L. Griffiths, "Evaluating Vector-Space Models of Word Representation, or, The Unreasonable Effectiveness of Counting Words near Other Words," *Proceedings of the 39th Annual Meeting of the Cognitive Science Society (CogSci 2017)*, London (G. Gunzelmann, A. Howes, T. Tenbrink and E. J. Davelaar, eds.), Seattle, WA: The Cognitive Science Society, pp. 859–864. api.semanticscholar.org/CorpusID:32843950.

- [32] A. Joulin, E. Grave, P. Bojanowski and T. Mikolov, “Bag of Tricks for Efficient Text Classification” in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL’17)*, Valencia, Spain (M. Lapata, P. Blunsom and A. Koller, eds.), Stroudsburg, PA: Association for Computational Linguistics, 2017 pp. 427–431. aclanthology.org/E17-2068.
- [33] J. Pennington, R. Socher and C. Manning, “GloVe: Global Vectors for Word Representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP’14)*, Doha, Qatar (A. Moschitti, B. Pang and W. Daelemans, eds.), Stroudsburg, PA: Association for Computational Linguistics, 2014 pp. 1532–1543. doi:10.3115/v1/D14-1162.
- [34] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee and L. Zettlemoyer, “Deep Contextualized Word Representations.” arxiv.org/abs/1802.05365.
- [35] J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1, Minneapolis, MN (J. Burstein, C. Doran and T. Solorio, eds.), Minneapolis, MN: Association for Computational Linguistics, 2019 pp. 4171–4186. doi:10.18653/v1/N19-1423.
- [36] F. Agbavor and H. Liang, “Predicting Dementia from Spontaneous Speech Using Large Language Models,” *PLoS Digital Health*, 1(12), 2022 e0000168. doi:10.1371/journal.pdig.0000168.
- [37] R. Al-Rfou’, B. Perozzi and S. Skiena, “Polyglot: Distributed Word Representations for Multilingual NLP,” in *Proceedings of the Seventeenth Conference on Computational Natural Language Learning (CoNLL’13)*, Sofia, Bulgaria (J. Hockenmaier and S. Riedel, eds.), Stroudsburg, PA: Association for Computational Linguistics, 2016 pp. 183–192. www.aclweb.org/anthology/W13-3520.
- [38] X. Zhou, X. Wan and J. Xiao, “Cross-Lingual Sentiment Classification with Bilingual Document Representation Learning,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL’16)*, Berlin, Germany (K. Erk and N. Smith, eds.), Stroudsburg, PA: Association for Computational Linguistics, 2016 pp. 1403–1412. doi:10.18653/v1/P16-1133.
- [39] B. Eisner, T. Rocktaschel, I. Augenstein, M. Bosnjak and S. Riedel, “emoji2vec: Learning Emoji Representations from Their Description,” in *Proceedings of the Fourth International Workshop on Natural Language Processing for Social Media(SocialNLP’16)*, Austin, TX (L. Ku, J. Hsu and C. Li, eds.), Stroudsburg, PA: Association for Computational Linguistics, 2016 pp. 48–54. doi:10.18653/v1/W16-6208.

- [40] D. Tang, F. Wei, B. Qin, N. Yang, T. Liu and M. Zhou, "Sentiment Embeddings with Applications to Sentiment Analysis," *IEEE Transactions on Knowledge and Data Engineering*, **28**(2), 2016 pp. 496–509. doi:10.1109/TKDE.2015.2489653.
- [41] O. Melamud, J. Goldberger and I. Dagan, "context2vec: Learning Generic Context Embedding with Bidirectional LSTM," in *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL'16)*, Berlin, Germany (S. Riezler and Y. Goldberg, eds.), Stroudsburg, PA: Association for Computational Linguistics, 2016 pp. 51–61. doi:10.18653/v1/K16-1006.
- [42] B. McCann, J. Bradbury, C. Xiong and R. Socher, "Learned in Translation: Contextualized Word Vectors," in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*, Long Beach, CA (U. von Luxburg, I. Guyon, S. Bengio, H. Wallach and R. Fergus, eds.), Red Hook, NY: Curran Associates Inc., 2017 pp. 6297–6308. dl.acm.org/doi/10.5555/3295222.3295377.
- [43] P. Xu, A. Madotto, C.-S. Wu, J. H. Park and P. Fung, "Emo2Vec: Learning Generalized Emotion Representation by Multi-task Training," in *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA'18)*, Brussels, Belgium (A. Balahur, S. M. Mohammad, V. Hoste and R. Klinger, eds.), Stroudsburg, PA: Association for Computational Linguistics, 2018 pp. 292–298. doi:10.18653/v1/W18-6243.
- [44] J. Howard and S. Ruder, "Universal Language Model Fine-Tuning for Text Classification," in *Proceedings of 56th Annual Meeting of the Association for Computational Linguistics (ACL'18)*, Melbourne, Australia (I. Gurevych and Y. Miyao, eds.), Stroudsburg, PA: Association for Computational Linguistics, 2018 pp. 328–339. doi:10.18653/v1/P18-1031.
- [45] C. Baziotis, A. Nikolaos, A. Chronopoulou, A. Kolovou, G. Paraskevopoulos, N. Ellinas, S. Narayanan and A. Potamianos, "NTUA-SLP at SemEval-2018 Task 1: Predicting Affective Content in Tweets with Deep Attentive RNNs and Transfer Learning," in *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval'18)*, New Orleans, LA (M. Apidianaki, S. M. Mohammad, J. May, E. Shutova, S. Bethard and M. Carpuat, eds.), Stroudsburg, PA: Association for Computational Linguistics, 2018 pp. 245–255. doi:10.18653/v1/S18-1037.
- [46] B. H. Soleimani and S. Matwin, "Spectral Word Embedding with Negative Sampling," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI'18)*, New Orleans, LA: AAAI Press, 2018 pp. 5481–5487. doi:10.1609/aaai.v32i1.12015.

- [47] S. Cao, W. Lu, J. Zhou and X. Li, “cw2vec: Learning Chinese Word Embeddings with Stroke n-gram Information,” in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI’18)*, New Orleans, LA, Palo Alto, CA: AAAI Press, 2018 pp. 5053–5061. doi:10.1609/aaai.v32i1.12029.
- [48] T. Wada, T. Iwata and Y. Matsumoto, “Unsupervised Multilingual Word Embedding with Limited Resources Using Neural Language Models,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL’19)*, Florence, Italy (A. Korhonen, D. Traum and L. Màrquez, eds.), Stroudsburg, PA: Association for Computational Linguistics, 2019 pp. 3113–3124. doi:10.18653/v1/P19-1300.
- [49] G. Lample and A. Conneau, “Cross-Lingual Language Model Pretaining.” arxiv.org/abs/1901.07291.
- [50] J. Armengol-Estapé, O. de Gibert Bonet and M. Melero, “On the Multilingual Capabilities of Very Large-Scale English Language Models,” in *Proceedings of the Thirteenth Language Resources and Evaluation Conference (LREC’22)*, Marseille, France: European Language Resources Association, 2022 pp. 3056–3068. aclanthology.org/2022.lrec-1.327.
- [51] J. Dobрева, T. Pavlov, K. Mishev, M. Simjanoska, S. Tudzarski, D. Trajanov and L. Kocarev, “MACEDONIZER - The Macedonian Transformer Language Model,” in *ICT Innovations 2022. Reshaping the Future Towards a New Normal (CCIS’22)*, Skopje, Macedonia (K. Zdravkova and L. Basnarkovand, eds.), Cham: Springer Nature Switzerland, 2022 pp. 51–62. doi:10.1007/978-3-031-22792-9_5.
- [52] P. Pal Chaudhuri, D. R. Chowdhury, S. Nandi and S. Chattopadhyay, *Additive Cellular Automata: Theory and Applications*, Los Alamitos, CA: IEEE Computer Society Press, 1997.
- [53] M. Elizabeth, S. M. Parsotambhai and R. Hazari, “Cellular Automata Enhanced Machine Learning Model for Toxic Text Classification,” in *Cellular Automata: 15th International Conference on Cellular Automata for Research and Industry (ACRI’22)*, Geneva, Switzerland (B. Chopard, S. Bandini, A. Dennunzio, M. A. Haddad, eds.), Cham: Springer, 2022 pp. 346–355. doi:10.1007/978-3-031-14926-9_31.
- [54] R. Hazari and S. Das, “Rule 136 and Its Equivalents Are Liberal, but Become Conservative in Their Conjugal Life,” *International Journal of Modern Physics C*, 29(06), 2018 1850040. doi:10.1142/S0129183118500407.
- [55] M. J. Elizabeth, A. K. Panda, P. P. Chaudhuri and R. Hazari, “Cellular Automata-Based Sentiment Analysis,” in *Proceedings of Second Asian Symposium on Cellular Automata Technology (ASCAT’23)*, Singapore (S. Das and G. J. Martinez, eds.), Singapore: Springer, 2023 pp. 53–64. doi:10.1007/978-981-99-0688-8_5.
- [56] S. Wolfram, ed., *Theory and Applications of Cellular Automata: Including Selected Papers 1983–1986*, Singapore: World Scientific, 1986.

- [57] G. A. Ruz, P. A. Henriquez and A. Mascareño, “Sentiment Analysis of Twitter Data during Critical Events through Bayesian Networks Classifiers,” *Future Generation Computer Systems*, **106**, 2020 pp. 92–104. doi:10.1016/j.future.2020.01.005.
- [58] M. Wozniak, M. Wiczorek and J. Silka, “BiLSTM Deep Neural Network Model for Imbalanced Medical Data of IoT Systems,” *Future Generation Computer Systems*, **141**, 2023 pp. 489–499. doi:10.1016/j.future.2022.12.004.
- [59] S. Minaee, E. Azimi and A. A. Abdolrashidi, “Deep-Sentiment: Sentiment Analysis Using Ensemble of CNN and Bi-LSTM Models.” arxiv.org/abs/1904.04206.
- [60] U. D. Gandhi, P. M. Kumar, G. C. Babu and G. Karthick, “Sentiment Analysis on Twitter Data by Using Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM),” *Wireless Personal Communications*, **2021**. doi:10.1007/s11277-021-08580-3.
- [61] J. Wang, L. C. Yu, K. R. Lai and X. Zhang, “Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL’16)*, Berlin, Germany (K. Erk and N. Smith, eds.), Stroudsburg, PA: Association for Computational Linguistics, 2016 pp. 225–230. doi: 10.18653/v1/P16-2037.
- [62] A. Yadav. “Restaurant Reviews | Kaggle.” (Sep 18, 2024) www.kaggle.com/code/aniketyadav1/restaurant-reviews.
- [63] S. Singh. “Sentiment Analysis Data | Kaggle.”
- [64] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng and C. Potts, “Recursive Deep Models for Semantic Compositionality over a Sentiment Treebank,” in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP’13)*, Seattle, WA (D. Yarowsky, T. Baldwin, A. Korhonen, K. Livescu and S. Bethard, eds.), Stroudsburg, PA: Association for Computational Linguistics, 2013 pp. 1631–1642. aclanthology.org/D13-1170.
- [65] M. Siddhartha and A. Bhatt. “Amazon Alexa Reviews | Kaggle.” (Sep 18, 2024) www.kaggle.com/datasets/sid321axn/amazon-alexa-reviews.
- [66] G. Gautam and D. Yadav, “Sentiment Analysis of Twitter Data Using Machine Learning Approaches and Semantic Analysis,” in *2014 Seventh International Conference on Contemporary Computing (IC3’14)*, Noida, India (M. Parashar, U. Bellur, S. D. M. Kumar, P. Chandran, M. Krishnan, K. Madduri, et al., eds.), Piscataway, NJ: IEEE, 2014 pp. 437–442. doi:10.1109/IC3.2014.6897213.
- [67] K. L. Tan, C. P. Lee, K. S. M. Anbananthen and K. M. Lim, “RoBERTa-LSTM: A Hybrid Model for Sentiment Analysis with Transformers and Recurrent Neural Network,” *IEEE Access*, **10**, 2022 pp. 21517–21525. doi:10.1109/ACCESS.2022.3152828.

- [68] J. Singh and P. Tripathi, "Sentiment Analysis of Twitter Data by Making Use of SVM, Random Forest and Decision Tree Algorithm," in *10th IEEE International Conference on Communication Systems and Network Technologies (CSNT'21)*, Bhopal, India (G. S. Tomar, ed.), Piscataway, NJ: IEEE, 2021 pp. 193–198.
doi:10.1109/CSNT51715.2021.9509679.
- [69] M. Govindarajan, "Sentiment Analysis of Restaurant Reviews Using Hybrid Classification Method," *International Journal of Soft Computing and Artificial Intelligence*, 2(1), 2014 pp. 17–23.
www.ijraset.com/files/serve.php?FID=34737.
- [70] M. Adnan, R. Sarno and K. R. Sungkono, "Sentiment Analysis of Restaurant Review with Classification Approach in the Decision Tree-J48 Algorithm," in *International Seminar on Application for Technology of Information and Communication (iSemantic'19)*, Semarang, Indonesia, Piscataway, NJ: IEEE, 2019 pp. 121–126.
doi:10.1109/ISEMANTIC.2019.8884282.
- [71] M. M. Hamad, M. A. Salih and R. A. Jaleel, "Sentiment Analysis of Restaurant Reviews in Social Media Using Naïve Bayes," *Applications of Modelling and Simulation*, 5, 2021 pp. 166–172.
arqiipubl.com/ojs/index.php/AMS_Journal/article/view/304.
- [72] S. Sazzed, "A Hybrid Approach of Opinion Mining and Comparative Linguistic Analysis of Restaurant Reviews," in *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP'21)*, Online (R. Mitkov and G. Angelova, eds.), INCOMA Ltd., 2021 pp. 1281–1288. aclanthology.org/2021.ranlp-1.144.
- [73] R. Tang, Y. Lu, L. Liu, L. Mou, O. Vechtomova and J. Lin, "Distilling Task-Specific Knowledge from BERT into Simple Neural Networks." arxiv.org/abs/1903.12136.
- [74] Q. Chen, R. Zhang, Y. Zheng and Y. Mao, "Dual Contrastive Learning: Text Classification via Label-Aware Data Augmentation." arxiv.org/abs/2201.08702.
- [75] M. E. Basiri, S. Nemati, M. Abdar, E. Cambria and U. R. Acharya, "ABCDM: An Attention-Based Bidirectional CNN-RNN Deep Model for Sentiment Analysis," *Future Generation Computer Systems*, 115, 2021 pp. 279–294. doi:10.1016/j.future.2020.08.005.
- [76] Z. Yuan, S. Wu, F. Wu, J. Liu and Y. Huang, "Domain Attention Model for Multi-domain Sentiment Classification," *Knowledge-Based Systems*, 155, 2018 pp. 1–10. doi:10.1016/j.knosys.2018.05.004.
- [77] Z. Jianqiang, G. Xiaolin and Z. Xuejun, "Deep Convolution Neural Networks for Twitter Sentiment Analysis," *IEEE Access*, 6, 2018 pp. 23253–23260. doi:10.1109/ACCESS.2017.2776930.
- [78] A. S. M. AlQahtani, "Product Sentiment Analysis for Amazon Reviews," *International Journal of Computer Science & Information Technology (IJCSIT)*, 13(3), 2021 pp. 15–30.
doi:10.5121/ijcsit.2021.13302.

- [79] S. A. Aljuhani and N. S. Alghamdi, “A Comparison of Sentiment Analysis Methods on Amazon Reviews of Mobile Phones,” *International Journal of Advanced Computer Science and Applications*, 10(6), 2019 pp. 608–617. doi:10.14569/IJACSA.2019.0100678.
- [80] R. A. Laksono, K. R. Sungkono, R. Sarno and C. S. Wahyuni, “Sentiment Analysis of Restaurant Customer Reviews on TripAdvisor Using Naïve Bayes,” in *Proceedings of 2019 International Conference on Information and Communication Technology and Systems (ICTS'19)*, Surabaya, Indonesia, Piscataway, NJ: IEEE, 2019 pp. 49–54. doi:10.1109/ICTS.2019.8850982.
- [81] R. Rajasekaran, U. Kanumuri, M. S. Kumar, S. Ramasubbareddy and S. Ashok, “Sentiment Analysis of Restaurant Reviews,” in *Smart Intelligent Computing and Applications: Proceedings of the Second International Conference on SCI 2018 (SIST'18)*, Singapore (S. C. Satapathy, V. Bhateja and S. Das, eds.): Singapore: Springer, 2018 pp. 383–390. doi:10.1007/978-981-13-1927-3_41.
- [82] J. Wang, Y. Zhang, L.-C. Yu and X. Zhang, “Contextual Sentiment Embeddings via Bi-directional GRU Language Model,” *Knowledge-Based Systems*, 235, 2022 107663. doi:10.1016/j.knosys.2021.107663.
- [83] A. Deniz, M. Angin and P. Angin, “Sentiment and Context-Refined Word Embeddings for Sentiment Analysis,” in *IEEE International Conference on Systems, Man, and Cybernetics (SMC'21)*, Melbourne, Australia, Piscataway, NJ: IEEE, 2021 pp. 927–932. doi:10.1109/SMC52423.2021.9659189.